



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/010,057	12/04/2001	Laurent P. Daynes	SUN-P6438-RSH	7842
22835	7590	03/25/2005	EXAMINER	
A. RICHARD PARK, REG. NO. 41241 PARK, VAUGHAN & FLEMING LLP 2820 FIFTH STREET DAVIS, CA 95616			TANG, KUO LIANG J	
		ART UNIT	PAPER NUMBER	
		2191		

DATE MAILED: 03/25/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/010,057	DAYNES ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	Kuo-Liang J Tang	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 03 November 2004.

2a) This action is **FINAL**.                                    2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 1-45 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 1-45 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All . b) Some \* c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.

4) Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.

5) Notice of Informal Patent Application (PTO-152)

6) Other: \_\_\_\_\_.

**DETAILED ACTION**

1. This Office Action is in response to the amendment filed on 11/3/2005.

The priority date for this application is 3/15/2001.

Claims 1, 16 and 31 are amended.

Claims 1-45 are pending and have been examined.

***Response to Arguments***

2. Applicant's arguments with respect to claims 1-45 have been considered but are moot in view of the new ground(s) of rejection.

Claims 1, 16 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's admitted prior art (hereinafter AAPA) further in view of Jardine et al., US Patent No. 6,195,754 (hereinafter Jardine).

Claims 2-5, 8-10, 17-20, 23-25, 32-35 and 38-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, in view of Jardine, further in view of Sowizral et al. US Patent No. 6,570,564 (hereinafter Sowizral) and further in view of Brundridge, US Patent No. 6,279,109.

Claims 6-7, 21-22 and 36-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, in view of Jardine, further in view of Sowizral, further in view of Brundridge, and further in view of Danforth, US Patent No. 6,085,034.

Claims 11-15, 26-30 and 41-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, in view of Jardine, further in view of Sowizral, further in view

of Brundridge, and further in view of Bak et al.,, US Patent No. 6,704,927 (hereinafter Bak).

*In the Amendment, the Applicants argue that:*

As for independent claims 1, 16 and 31, the Applicants primarily argue that Applicants' admitted prior art (AAPA) does not disclose "by setting the initializer field to indicate an event that triggered initialization". (see Amendment pages 16-17).

**Examiner's response:**

It should be noted below that Jardine teaches as such new amended claimed limitations and not the AAPA. And as setforth below the combination of AAPA and Jardine discloses "by setting the initializer field to indicate an event that triggered initialization" (E.g. see Jardine at col. 15:39-48, which states "...an I/O subsystem can use these events as a trigger for its initialization process ...") (emphasis added).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1, 16 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's admitted prior art (hereinafter AAPA) further in view of Jardine et al., US Patent No. 6,195,754 (hereinafter Jardine).

As Per Claim 1, AAPA teaches:

“augmenting the shared runtime data structure ((E.g. see Spec Page 3, lines 23, which states “... class of Java programming language ...”) representing a shared part of a class with an initializer field (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”; and

“using the initializer field of the class (E.g. see Spec Page 3, lines 23-24, which states “...four such instructions: getstatic, putstatic, invokestatic, new ...”) to determine whether a platform-independent (E.g. see Spec Page 3, line 22) instruction of the program method may trigger an initialization of the class (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2 and see Spec Page 4, lines 7-12)”.

AAPA does not explicitly disclose an event that triggered initialization. However, Jardine in an analogous art teaches “by setting the initializer field to indicate an event that triggered initialization”. (E.g. see col. 15:39-48, which states “...an I/O subsystem can use these events as a trigger for its initialization process ...”) (emphasis added). Therefore, it would have been obvious to incorporate the teaching of Jardine into the teaching of AAPA so that an event that triggered initialization. The modification would have been obvious because one of ordinary skill in the art would have been motivated to use events as a trigger for its initialization process to download software when power restored.

As Per Claim 16, is the computer-readable medium claim corresponding to the method claim 1 and is rejected under the same reason set forth in connection of the rejection of claim 1.

As Per Claim 31, is the apparatus claim corresponding to the method claim 1 and is rejected under the same reason set forth in connection of the rejection of claim 1.

4. Claims 2-5, 8-10, 17-20, 23-25, 32-35 and 38-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, in view of Jardine, further in view of Sowizral et al. US Patent No. 6,570,564 (hereinafter Sowizral) and further in view of Brundridge, US Patent No. 6,279,109.

As Per claim 2, the rejection of claim 1 is incorporated and further AAPA does not explicitly disclose initializing a bootstrap class. However, Sowizral in an analogous art teaches “initializing a bootstrap class”. (E.g. see Sowizral col. 15:21-31, which states “...to bootstrap itself ... initializer method may be included in the class ...”). Therefore, it would have been obvious to incorporate the teaching of Sowizral into the teaching of AAPA to initialize a bootstrap class. The modification would have been obvious because one of ordinary skill in the art would have been motivated to initialize class during system startup time.

The combination teaching of AAPA, Jardine and Sowizral does not explicitly disclose bootstrap class is initialized during startup of a task and before any creation of multiple threads. However, Brundridge in an analogous art teaches “bootstrap class is initialized during startup of a task of the multitasking virtual machine, and before any concurrency, due to creation of multiple threads of control within the task, takes place”. (E.g. see Brundridge col. 7:47-52). Therefore, it would have been obvious to incorporate

the teaching of Brundridge into the combination teaching of AAPA, Jardine and Sowizral so that bootstrap class is initialized during startup of a task and before any creation of multiple threads. The modification would have been obvious because one of ordinary skill in the art would have been motivated so that a multi-threaded system can perform multiple operations at one time without waiting for another process to complete.

As Per claim 3, the rejection of claim 2 is incorporated and further AAPA teaches:

“assigning a value of an initializer of the class when the class is fully initialized (E.g. see Spec Page 3, lines 23-24, which states “...four such instructions: getstatic, putstatic, invokestatic, new ...”, the assigned value is inherent after the method/function is executed and a value is returned to be assigned”;

“the class is the class and the value of the initializer further indicates that the class initialization was not triggered by a class initialization barrier (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”.

The combination teaching of AAPA and Jardine does not explicitly disclose initializing a bootstrap class. However, Sowizral in an analogous art teaches “initializing a bootstrap class”. (E.g. see Sowizral col. 15:21-31, which states “...to bootstrap itself ... initializer method may be included in the class ...”). Therefore, it would have been obvious to incorporate the teaching of Sowizral into the teaching of AAPA and Jardine to initialize a bootstrap class. The modification would have been obvious because one of ordinary skill in the art would have been motivated to initialize class during system startup time.

As Per claim 4, the rejection of claim 3 is incorporated and further further AAPA and Jardine do not explicitly disclose initializing a bootstrap class. However, Sowizral in an analogous art teaches “initializing a bootstrap class”. (E.g. see Sowizral col. 15:21-31, which states “...to bootstrap itself ... initializer method may be included in the class ...”). The combination teaching of AAPA, Jardine and Sowizral teaches:

“setting a binary variable to zero upon starting the multitasking virtual machine (Again see as noted above of Claim 2, Sowizral teaches “initializing a bootstrap class”( E.g. see Sowizral col. 15:21-31, which states “...to bootstrap itself ... initializer method may be included in the class ...”). Therefore it is inherent that at the beginning of the class initializer, a default value is set (i.e. NULL, in binary format 0x00) if it is not specific specified.”; and

“setting the binary variable to one when all bootstrap classes have been initialized by a first task executed by the multitasking virtual machine (Again see as noted above of Claim 3, a returned value will be assigned)”;

“whereby the binary variable indicates to the multitasking virtual machine whether all bootstrap classes have been initialized (Again see as noted above of Claim 3, a returned value differe than the default will be the indication that the class is fully initialized).”

As Per claim 5, the rejection of claim 4 is incorporated and further further The combination teaching of AAPA, Jardine and Sowizral teaches:

“upon initiating the initialization of the class from a class initialization barrier, noting the holder of the class initialization barrier; and once the class is fully initialized, assigning the holder to the initializer field only if the binary variable is zero(Again, see as noted above of Claim 4)”.

As Per claim 8, the rejection of claim 3 is incorporated and further the AAPA teaches “wherein a pointer to a runtime data structure representing the shared part of the class is assigned to the initializer field of the class to indicate that the class is initialization is not triggered by a class initialization barrier (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”. AAPA and Jardine do not explicitly disclose initializing a bootstrap class. However, Sowizral in an analogous art teaches “initializing a bootstrap class”. (E.g. see Sowizral col. 15:21-31, which states “...to bootstrap itself ... initializer method may be included in the class ...”). Therefore, it would have been obvious to incorporate the teaching of Sowizral into the teaching of AAPA and Jardine to initialize a bootstrap class. The modification would have been obvious because one of ordinary skill in the art would have been motivated to initialize class during system startup time.

As Per claim 9, the rejection of claim 3 is incorporated and further the AAPA teaches:

“if the class does not have an initialization sequence, setting the class to a fully initialized state upon the class being loading by the task, and assigning the initializer field of the class to a pointer to a runtime data structure representing the shared part of the class (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”.

As Per claim 10, the rejection of claim 3 is incorporated and further the AAPA teaches:

“instructing the dynamic compiler not to generate native code for the class initialization barrier of the program method being compiled if the class targeted by the class initialization barrier is equal to the class that defines the program method being compiled (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”.

As per Claims 17-20 and 23-25, the rejection of claim 16 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 2-5 and 8-10 respectfully.

As per Claims 32-35 and 38-40, the rejection of claim 31 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 2-5 and 8-10 respectfully.

5. Claims 6-7, 21-22 and 36-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, in view of Jardine, further in view of Sowizral, further in view of Brundridge, and further in view of Danforth, US Patent No. 6,085,034.

As Per claim 6, the rejection of claim 3 is incorporated and further the combination teaching of AAPA, Jardine, Sowizral and Brundridge does not explicitly disclose upon setting a non-bootstrap class to a fully initialized state for a task, assigning

the initializer field of the class to a constant value, wherein the constant value is distinguishable from all other possible values for the initializer field. However, Danforth in an analogous art teaches “upon setting a non-bootstrap class to a fully initialized state for a task, assigning the initializer field of the class to a constant value, wherein the constant value is distinguishable from all other possible values for the initializer field”. (E.g. see Danforth FIG. 6, step 305 and associated text)”. Therefore, it would have been obvious to incorporate the teaching of Danforth into the combination teaching of AAPA, Jardine, Sowizral and Brundridge to initialize a non-bootstrap class. The modification would have been obvious because one of ordinary skill in the art would have been motivated to increase the optimization of the system by initialize non-bootstrap class during system startup time.

As Per claim 7, the rejection of claim 6 is incorporated and further the combination teaching of AAPA, Jardine, Sowizral and Brundridge and Danforth teaches: “wherein the constant value is a NULL pointer” (E.g. see Danforth FIG. 6, step 303 and associated text).

As per Claims 21-22, the rejection of claim 18 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 6-7 respectfully.

As per Claims 36-37, the rejection of claim 33 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 6-7 respectfully.

6. Claims 11-15, 26-30 and 41-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over AAPA, in view of Jardine, further in view of Sowizral, further in view of Brundridge, and further in view of Bak et al., US Patent No. 6,704,927 (hereinafter Bak).

As Per claim 11, the rejection of claim 10 is incorporated and further the AAPA and Jardine teach “instructing the dynamic compiler not to generate native code for the class initialization barrier of the program method being compiled if the class targeted by the class initialization barrier is a class of the class that defines the program method being compiled (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”.

The combination teaching of AAPA, Jardine, Sowizral and Brundridge does not explicitly super class. However, Bak in an analogous art teaches “instructing the dynamic compiler not to generate native code for the class initialization barrier of the program method being compiled if the class targeted by the class initialization barrier is a class of the superclass (E.g. see FIG. 8 and associated text, e.g. see col. 2:39-58) that defines the program method being compiled. Therefore, it would have been obvious to incorporate the teaching of Sowizral into the teaching of AAPA and Jardine to initialize a bootstrap class. The modification would have been obvious because one of ordinary skill in the art would have been motivated to analyze a first class associated with a class hierarchy of a system during run-time includes marking the first class and marking a second class that is a superclass of the first class to indicate an associated between the two class.

As Per claim 12, the rejection of claim 11 is incorporated and further the AAPA teaches:

“wherein the value of the holder of the class initialization barrier is a pointer to a runtime data structure representing the shared part of the class that defines the program method that holds the class initialization barrier” (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2).

As Per claim 13, the rejection of claim 12 is incorporated and further the AAPA teaches:

“instructing the dynamic compiler not to generate native code for the class initialization barrier of the program method being compiled if the value of the initializer field of the class targeted by the class initialization barrier is: different from the value that indicates that the class is not bootstrap class (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2), and different from the pointer to the runtime data structure representing the shared part of the class that defines the program method being compiled (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2).

As Per claim 14, the rejection of claim 11 is incorporated and further the AAPA teaches:

“wherein the value of the holder of the class initialization barrier is a pointer to the shared runtime data structure representing the program method that holds the class initialization barrier (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”.

As Per claim 15, the rejection of claim 14 is incorporated and further the AAPA teaches:

“instructing the dynamic compiler not to generate native code for the class initialization barrier of the program method being compiled if the value of the initializer field of the class targeted by the class initialization barrier is: different from the value that indicate that the class is not the bootstrap class, and different from the pointer to the shared runtime data structure representing the program method being compiled (E.g. see Spec Page 3, lines 19-26 to Page 4, lines 1-2)”.

As per Claims 26-30, the rejection of claim 25 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 11-15 respectfully.

As per Claims 41-45, the rejection of claim 40 are incorporated and are rejected under the same reason set forth in connection of the rejection of claims 11-15 respectfully.

### *Conclusion*

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

***Correspondence Information***

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuo-Liang J Tang whose telephone number is (571) 272-3705. The examiner can normally be reached on 8:30AM - 7:00PM (Monday – Thursday).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Software Engineer Patent Examiner



TUAN DAM  
SUPERVISORY PATENT EXAMINER